Report No. NAWCADWAR-95003-4.5

AN ANALYSIS OF GAUSS ELIMINATION FOR ADAPTIVE BEAMFORMING

Peter R. Turner, Ph.D.
Mathematics Department
U.S. NAVAL ACADEMY
Annapolis, MD 21402

Barry J. Kirsch
Avionics Department
Engineering Division (Code 4.5.5.1)
NAVAL AIR WARFARE CENTER
AIRCRAFT DIVISION WARMINSTER
P.O. Box 5152
Warminster, PA 18974-0591

DTIC
SELECTED
SEP 2 7 1995
F

9 AUGUST 1994

19950926 150

FINAL REPORT

NOTICES

REPORT NUMBERING SYSTEM - The numbering of technical project reports issued by the Naval Air Warfare Center, Aircraft Division, Warminster is arranged for specific identification purposes. Each number consists of the Center acronym, the calendar year in which the number was assigned, the sequence number of the report within the specific calendar year, and the official 2-digit correspondence code of the Functional Department responsible for the report. For example: Report No. NAWCADWAR-95010-4.6 indicates the tenth Center report for the year 1995 and prepared by the Crew Systems Engineering Department. The numerical codes are as follows.

Code	Department
4.1	Systems Engineering Department
4.2	Cost Analysis Department
4.3	Air Vehicle Department
4.4	Propulsion and Power Department
4.5	Avionics Department
4.6	Crew Systems Engineering Department
4.10	Conc. Analy., Eval. and Plan (CAEP) Department

PRODUCT ENDORSEMENT - The discussion or instructions concerning commercial products herein do not constitute an endorsement by the Government nor do they convey or imply the license or right to use such products.

Reviewed By:___*Barry Kirsch*___ Date:_2/16/95_
Author/COTR

Reviewed By:_____ Date:_4/4/5_
LEVEL III Manager

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 9 AUGUST 94	3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE

AN ANALYSIS OF GAUSS ELIMINATION FOR ADAPTIVE BEAMFORMING

5. FUNDING NUMBERS

6. AUTHOR(S)

*PETER R. TURNER, PH.D., and BARRY J. KIRSCH**

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

** Avionics Department; Engineering Division (Code 4.5.5.1)
NAVAL AIR WARFARE CENTER; AIRCRAFT DIVISION WARMINSTER
P.O. Box 5152
Warminster, PA 18974-0591

8. PERFORMING ORGANIZATION REPORT NUMBER

NAWCADWAR-95003-4.5

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr, Sherman Gee
OFFICE OF NAVAL RESEARCH
ONR-313
800 N. Quincy Street, Arlington, VA 22217

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

* Mathematics Department
U.S. NAVAL ACADEMY
Annapolis, MD 21402

12a. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

There are many formulations of the Adaptive Beamforming (ABF) problem. We are motivated to use an algorithm that uses the least number of divisions or square-root operations, since these operations are very expensive in time or chip area. One method solves the solution of a Hermitian matrix. The Gauss Elimination (GE) algorithm has no square-roots. In this paper, we analyze dynamic range requirements and precision analysis for an integer processor implementation of GE (as opposed to fixed-point or floating-point), to solve the ABF problem. This analysis differs from the standard Wilkinson analysis which is based on fixed-point arithmetic, where the multiplier is formed once per row, whereas the integer formulation requires that we perform explicit divisions on each element of the reduced matrix. We present formulas to determine input quantization, storage wordlengths and accumulator wordlengths for a desired weight precision.

14. SUBJECT TERMS

ADAPTIVE BEAMFORMING (ABF), GAUSS ELIMINATION (GE)

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

An Analysis of Gauss Elimination for Adaptive Beamforming

Peter R Turner   and   Barry J Kirsch

Mathematics Department   Code 5051

U S Naval Academy   NAWC - Aircraft Division

Annapolis, MD 21402   Warminster, PA 18974

prt@sma.usna.navy.mil   bkirsch@nadc.nadc.navy.mil

Introduction and Motivation:

We have been studying [6], [7] an integer-only arithmetic called the Residue Number System. The consequences of RNS being an integer-only arithmetic include the inability to do division, and square-root operations. The applications that are very amenable to integer-only arithmetic, are the basic signal processing functions that are multiply-accumulate intensive, such as FIR (Finite Impulse Response) digital filters, convolution, correlation, FFTs (Fast Fourier Transforms). These operations have no feedback and hence the required dynamic range is finite, and can easily be predicted for worst case growth. The processor required to perform these functions must be able to handle the worst case growth. If it is expected that the computed results will exceed the processors dynamic range (either before processing starts or continually checked during processing), scaling will be required so that overflow will be avoided. Overflow will result in unrecoverable errors in the output yielding meaningless solutions.

In past research, we have found, through simulation and analysis, that the Residue Number System arithmetic can provide up to 8 times speed improvement over a state-of-the-art DSP (digital signal processor), based on the conventional binary integer arithmetic, at the same clock speed. This improvement is based on multiply-accumulate-only operations. No division is considered. If division is part of the algorithm, the improvement is less, due to the overhead of RNS-to-Binary conversion. For comparison purposes, it was assumed that the non-RNS operations were performed on the same processor as the binary processor under comparison, so the degradation is mostly due to the RNS-to-Binary conversion overhead.

Scaling operations must be handled very carefully. There are a few general issues that must be examined. First, does the problem allow scaling, and must the scale factor be saved, to recover the true solution, or can the scale factor be ignored. For example, in the adaptive beamforming problem, $R\mathbf{w} = \mathbf{s}$, the weight vector may be scaled by any real number since it is only the *relative* magnitudes of the weights that are of importance. In this example, the scale factor does not need to be saved. The problem though, is that too much scaling will reduce the precision of the solution, resulting in degraded overall signal-to-noise ratio.

Another issue is defining the difference between the scale operation and division, and the choice of the scale factor to be used. We define scaling as dividing by an arbitrary number, to decrease the required dynamic range to store the result. We use the term division, when the divisor is not arbitrary, but is a required for a particular algorithm. For example, in linear algebra, there are algorithms that require a normalized vector $\mathbf{v}/\|\mathbf{v}\|$. This division by the norm of the vector is used, for example, in producing sets of orthonormal vectors. If the division were replaced by an arbitrary scale factor, the algorithm may not necessarily produce orthogonal vectors, which may exacerbate any ill-conditioning problems.

In terms of speed, the scaling operation is more desirable than the division operation. For instance, in a binary computer, scaling (division by 2^n) is done by shifting the bits of a word to the right, n times - a simple operation in binary. Division by a specific number requires many more operations, including shifts and subtracts. In the residue number system, division requires conversion of the RNS representation back to binary using the Chinese Remainder Theorem, CRT. (See [11], for example.) The quotient is then rounded to an integer and converted back to the RNS. The RNS-to-Binary conversion is a very time consuming process though Binary-to-RNS conversion is more straightforward. On the other hand, like binary, scaling is a little simpler in RNS. There are a variety of methods. One method can scale the number by one or more of the moduli in the residue number system. Another method based on the CRT, called the L-CRT, developed by researchers at the University of Florida [4]. The L-CRT scales the RNS number by a fixed, known constant, and avoids a full RNS-to-Binary conversion.

Other researchers, including Westinghouse, have indicated that one of the biggest problems with

implementing Gauss Elimination in RNS, is the dynamic range issue and implementing scaling to overcome it. The mathematical details, including accuracy tradeoffs, were never presented. We will specifically address this issue.

The following analysis is to be used as background for the ultimate questions of whether we can use an RNS processor successfully for adaptive beamforming, ABF, applications, what constraints are required for the processor, and guidelines to design a processor for adaptive beamforming. We will study, in detail, the dynamic range requirements, for the Gauss Elimination algorithm, which can be used for adaptive beamforming. We chose to study Gauss Elimination because of its simplicity in implementation. Though there are other more robust algorithms that should be studied later, we observe that for the covariance matrix formulation of the ABF problem, the matrix is positive definite Hermitian so that there is no significant theoretical advantage to other methods. The QR algorithm for the least squares formulation of the problem is numerically stable but in its basic form requires substantial use of non-RNS operations such as square-roots. This approach has been studied in terms of architecture and some precision analysis by Ward, Hargrave and McWhirter [12].

For completeness it is desirable to say a little here about the adaptive beamforming problem. A typical beamforming situation is shown in Figure 2. An array of N antenna elements are sampled at time k to form a complex snapshot vector $\mathbf{x}_k$. A collection of K of these snapshots constitute the $N \times K$ ($N < K$) data matrix X. Inner products between the data vector $\mathbf{x}_k$ and complex weights $\mathbf{w}$ form the complex scalar outputs y_k. For the time from 1 to K, the output vector $\mathbf{y} = \mathbf{w}^H X$. The problem is to determine the weights $w_0, w_1, ..., w_{N-1}$ that will optimize the response $\mathbf{y}$ in some sense. When it is necessary to continually adjust the weights, we say that we are doing *adaptive beamforming*.

Thus we have

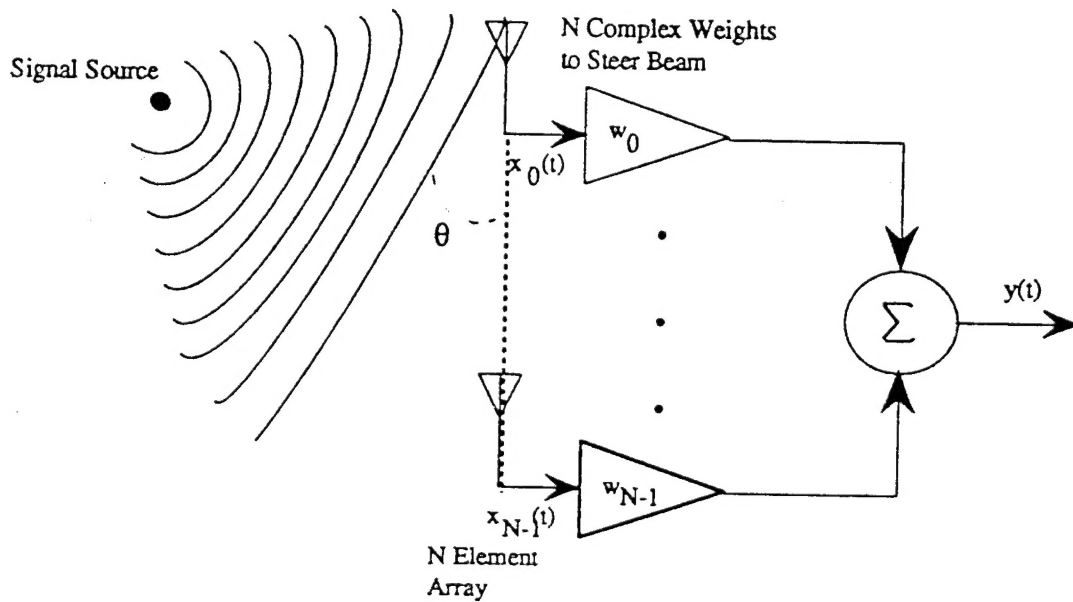$$\text{Input} = \mathbf{x}(t) = \begin{pmatrix} x_0(t) \\ x_1(t) \\ \vdots \\ x_{N-1}(t) \end{pmatrix}$$

and seek

$$\text{Weights} = \mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N-1} \end{pmatrix}$$

The situation is illustrated in Figure 1 below.

FIGURE 1

We can derive the optimal weights to minimize the mean-square error, MSE = $E[\epsilon^2]$, where the error signal, ϵ is the difference between the desired response and the output y.

$$\epsilon_k = d_k - y_k = d_k - \mathbf{w}^H \mathbf{x}_k$$
$$\epsilon_k^2 = d_k^2 - 2d_k \mathbf{w}^H \mathbf{x}_k + \mathbf{w}^H \mathbf{x}_k \mathbf{x}_k^H \mathbf{w}$$

Taking expected values of both sides yields

$$E[\epsilon^2] = \overline{\epsilon_k^2} = \overline{d_k^2} - 2\mathbf{w}^H \overline{\mathbf{x}_k d_k} + \mathbf{w}^H \overline{\mathbf{x}_k \mathbf{x}_k^H} \mathbf{w}$$

or

$$E[\epsilon^2] = \overline{d_k^2} - 2\mathbf{w}^H \mathbf{r}_{xd} + \mathbf{w}^H R_{xx} \mathbf{w}$$

To minimize this function, we set the gradient with respect to the weight vector equal to zero, that is,

$$\nabla \epsilon^2 = -2\mathbf{r}_{xd} + 2R_{xx}\mathbf{w} = 0 \quad \rightarrow \quad R_{xx}\mathbf{w} = \mathbf{r}_{xd}$$

An approximation R to the correlation matrix R_{xx} (also called the covariance matrix for zero-mean data [8]) is formed from the $N \times K$ data matrix X. R_{xx} is the complex $N \times N$ matrix $R_{xx} = E[XX^H]$ which is an infinite time average. Since we only have a finite number K of snapshots, we use the estimated covariance matrix

$$\hat{R} = XX^H / K$$

The covariance matrix is always non-singular, and hence $\hat{R}$ is a positive definite Hermitian matrix, since statistically independent noise exists on the antenna elements. The noise correlation matrix is just $R_n = \sigma^2 I$, where σ^2 is the noise variance (power), and I is the identity matrix of size N. That is, the cross-correlation terms average out while the autocorrelation terms average to the variance of the noise. The data covariance matrix is made from the sum of the signal, jammer and noise covariance matrices: $R = R_s + R_j + R_n$.

The weight vector is found by solving the system $R\mathbf{w} = \mathbf{s}$ where *either*

(a) $\mathbf{s}$ could be the *steering vector* given by

$$\mathbf{s} = \left(1, e^{-j\phi}, e^{-2j\phi}, ..., e^{-(N-1)j\phi} \right)^T$$

where $\phi = (2\pi d/\lambda)\sin\theta$ and θ is the desired look-angle with respect to the normal to the linear antenna array; d is the inter-element spacing and λ is the wavelength of the incoming signal at the carrier frequency, *or*

(b) $\mathbf{s}$ could be the *cross-correlation* vector

$$\mathbf{r}_{xd} = E[\mathbf{x}_k d_k^*] \approx (X\mathbf{d}^H) / K$$

where d_k is the reference signal sampled at time k, $\mathbf{x}_k = (x_0, x_1, ..., x_{N-1})^T$ is the snapshot vector at time k, and, as before, $E[\bullet]$ is the expectation operator.

Algorithms which have been used for solving this covariance matrix form of the problem include Gauss elimination, Cholesky decomposition, and the recursive least-squares (RLS) method based

on the matrix inverse lemma [5], p.385].

Parameters

Principal parameters of the problem and the notation we shall use for them are

N = # array elements

K = # "snapshots

q_w = # bits accuracy required for weight vector

q_X = # bits accuracy in data matrix, or "quantization" (also gives numerical range of data matrix entries)

κ = condition number of covariance matrix

$1+L_S$ = wordlength (in bits) for storage - equivalent to specifying dynamic range

L_A = accumulator wordlength.

The basic steps to analyse are:

1.   Formation of the covariance matrix

2.   Forward elimination

3.   Back substitution

4.   Sensitivity of the solution to q_w.

The analysis must consider both range and precision. Both of these will show some dependence on the eigenvalue structure and conditioning of the covariance matrix.

1.   Formation of the covariance matrix and cross-correlation vector

Each element of the covariance matrix is obtained as the inner product of two complex K-vectors with components in the interval $\left[-2^{q_x}, 2^{q_x}\right]$ so that the real and imaginary components of each complex product are in the range $\left[-2(2^{q_x})^2, 2(2^{q_x})^2\right] = \left[-2^{2q_x-1}, 2^{2q_x-1}\right]$ which in turn implies that real and imaginary parts of the elements of the scaled[1] covariance matrix lie in the interval $\left[-K \times 2^{2q_x-1}, K \times 2^{2q_x-1}\right]$.

If the full integer (scaled) covariance matrix is to be stored then it follows that

---

[1]   Note that the term "scaled" means "scaled up" throughout. The scaling results from the omission of the division by K.

$$L_S \geq 1 + 2q_X + \log K \tag{1}$$

Here and throughout log is used for the base-2 logarithm $\log_2$. Of course there will be much greater growth in the dynamic range (and so in the required value of L_S) during subsequent stages.

If, on the other hand, the actual covariance matrix is to be formed there is a division of these inner products by K so that the range is reduced by this factor and (1) is replaced by the requirement

$$L_S \geq 1 + 2q_X \tag{2}$$

Note that in the case where K is a power of 2, $K = 2^p$, this particular division is equivalent to using the final p bits of the accumulated sums for rounding of the L_S-bit words. In this case there is an error in the representation which for a sufficiently large accumulator can therefore be kept to a single roundoff in the final division. Specifically, if

$$L_A \geq 1 + 2q_X + \log K \tag{3}$$

then the absolute errors in the real and imaginary parts of the elements of the covariance matrix R are bounded by $1/2$.

We recall here some of the basic definitions of the matrix and vector norms which we are using in this discussion. Although the dynamic range is directly related to the magnitudes of the real and imaginary parts of the various complex quantities the analysis is simplified by using the magnitudes of the complex numbers themselves:

$$|z| = |x + jy| = \sqrt{x^2 + y^2} = \sqrt{zz^*} \tag{4}$$

For a complex vector $\mathbf{z} = (z_1, z_2, ..., z_n)^T$ the *maximum* or ∞-norm is defined by

$$\|\mathbf{z}\|_\infty = \max_i |z_i| \tag{5}$$

and then the associated matrix norm is defined as usual by

$$\|A\|_\infty = \max\{\|A\mathbf{z}\|_\infty : \|\mathbf{z}\|_\infty \leq 1\} \tag{6}$$

which is given by the maximum (absolute) row sum of the matrix A:

$$\|A\|_\infty = \max_i \sum_j |a_{ij}| \tag{7}$$

It follows now that the absolute errors in the elements of the computed covariance matrix R are bounded by $1/\sqrt{2}$; that is, the ∞-norm error bound for R is

$$\| \delta R \|_\infty \leq N/\sqrt{2} \tag{8}$$

Here we have used δR to denote the error in the matrix R; corresponding notation will be used for other errors subsequently.

The two cases - scaled and unscaled - must both be pursued for their effects on dynamic range and accuracy requirements. The bound (8) can be used to obtain error estimates for the solution obtained by Gauss elimination.

The formation of the cross-correlation vector **y** for the right-hand side is similar, with components consisting of inner products between the snapshot vectors comprising the data matrix and some desired response **d**. If we assume the same quantization for **d** as for the data matrix, then (1) and (3) summarize the options for the wordlength in just the same way as above. In the absence of scaling, the division by K again introduces an error which (using the same analysis as above) satisfies

$$\| \delta \mathbf{y} \|_\infty \leq 1/\sqrt{2} \tag{9}$$

Example   For the special case $N = 4$, $K = 16$, the inequality (1) becomes $L_s \geq 5+2q_X$ for the scaled matrix. Without the scaling $L_A \geq 5+2q_X$, and $L_s \geq 1+2q_X$, the error bound (8) is $\| \delta R \|_\infty \leq 2\sqrt{2}$.

The error bounds (8) and (9) can be used to obtain bounds on the error in the solution of the resulting (unscaled) linear system making use of condition number estimates which can be obtained from signal strengths [1]. The next short section summarizes this work on the eigenvalues of the covariance matrix.

2.   **The eigenvalue spread for the covariance matrix of adaptive beamforming**

Compton ([1] Section 4.6, pp258-275) studies the eigenvalues and therefore the condition number of the covariance matrix. His overall findings can be summarized as follows:

The eigenvalues of the covariance matrix (normalized relative to the background noise level) are all greater than or equal to unity so that

$$\lambda_{min} \geq 1 \tag{10}$$

The only non-unit eigenvalues are directly related to the powers of the various signals - both the

desired and interference signals. In the case of a single jammer of significantly greater power than the desired signal, the two largest eigenvalues are (approximately) proportional to the number of antennas and the powers of the jammer and of the desired signal respectively.

It follows that the condition number is well-approximated by N times the most powerful signal's (the jammer's) SNR. That is, with the same normalization as above:

$$\lambda_{max} \approx N \times SNR_J \qquad\qquad (11)$$

It follows that

$$\kappa \approx N \times SNR_J \qquad\qquad (12)$$

which for a jammer of 40dB with $N = 4$ antennas means that $\kappa \approx 4 \times 10^4$.

In the conventional error analysis for Gaussian elimination of Wilkinson [14] this condition number is essentially the scale factor by which errors are magnified during the solution. However the more recent analysis of Demmel [2] and others using a relative ∞-norm suggests that the norm of the inverse matrix may be a better condition number; and, in this case, that is close to unity.

The relative merits of these two analyses for the adaptive beamforming problem should be investigated further.

Monzingo and Miller [8] consider the effect of the eigenvalue spread on the accuracy of the solution obtained from the covariance matrix solution. In particular, their experimental analysis shows that with an eigenvalue spread of around 40 dB and using a mere 10 bit wordlength there was a degradation in the solution of no more than 2 dB which corresponds to only one or two significant *bits* rather than the four significant *decimal* figures which might be expected from the large condition number.

The other relevant aspect of their work to the present discussion centers around the question of the number of snapshots which are necessary in order to achieve acceptable accuracy in the solution. Their conclusion was that, for a 3 dB loss, $K = 2N$ was sufficient. It is anticipated that this figure is too small for many scenarios and therefore results here are presented for three cases: $N/K = 2, 3, 4$.

3.   Forward elimination

In this section we are primarily interested in the dynamic range requirements of the forward elimination phase of our solution process. We shall consider the solution of a linear system

$$A\mathbf{x} = \mathbf{b} \tag{13}$$

and denote the elements of the positive definite Hermitian matrix A by a_{ij} and the components of the right-hand-side vector by b_i. The basic algorithm for the forward elimination phase of Gauss elimination can then be written in the *ijk*-form as:

Forward elimination algorithm

> for $i = 1$ to $N-1$
> > for $j = i+1$ to N
> > > $m := a_{ji} / a_{ii}$
> > > $a_{ji} := 0$
> > > $b_j := b_j - mb_i$
> > > for $k = i+1$ to N
> > > > $a_{jk} := a_{jk} - ma_{ik}$

Again there are two cases to consider depending on whether the covariance matrix is or is not scaled.

3.1   The scaled covariance matrix

We assume here that (1) is satisfied and so the scaled (that is, no division by K) form of the covariance matrix can be used. Consider first the dynamic range growth which is implicit in the Gauss elimination phase of the solution. Since the covariance matrix is known to be positive definite Hermitian, no pivoting is used. (See [14], for example, for justification of this.)

At each step of the innermost loop of the process, a complex 2×2 matrix is being reduced as follows:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow \begin{bmatrix} a & b \\ 0 & d - bc/a \end{bmatrix} \tag{14}$$

The most immediate question is whether the division can be incorporated into the algorithm in a meaningful way. Now for our positive definite Hermitian matrix, the diagonal elements are all real and remain real throughout the elimination phase. Therefore the divisor a in (14) is always real and, for our problem and solution, must be a real integer.

The standard implementation of Gauss elimination as in the above algorithm would compute the

appropriate multiplier for the current row and then to use this multiplier for all the relevant matrix entries. In the integer arithmetic situation this is not possible since the multiplier would need to be rounded to a (complex) integer and this would always round to zero if the off-diagonal entry was smaller than the pivot. Since the largest element of the matrix necessarily lies on the diagonal [3], [13] this rounding of the multiplier to zero must occur at some point in the elimination phase.

The preferred alternative is therefore to compute $b \times c$ first in a full length accumulator and then to perform the rounded division by a. This places a constraint on the accumulator size in order to accommodate the full components of each complex multiplication:

$$L_A \geq 2L_S + 1 \tag{15}$$

The division by a of course reduces the range of this product. Wilkinson's analysis [14] of Gaussian elimination shows that for a positive definite matrix the growth factor is 1 and therefore that the true value of $d - bc/a$ in (14) is bounded by the largest element of the original matrix. The "growth factor" referred to here is the estimate of the ratio of the magnitudes of the largest elements in the original matrix and the final upper triangular matrix at the end of the elimination phase.

Now the division can be performed so as to give an integer result which has an error no greater than 1/2 and *provided the positive definiteness is not lost* during the elimination phase the only element growth that can occur results from these rounding errors so that

$$\max |a_{ij}^{(N)}|_\infty \leq \max |a_{ij}^{(1)}|_\infty + (N-1)/2 \tag{16}$$

where the $a_{ij}^{(k)}$ are the elements of the matrix at the kth stage of the elimination and $|\bullet|_\infty$ denotes the magnitude of the largest component of its complex argument. That is

$$|z|_\infty = \max \{ |\text{Re}(z)|, |\text{Im}(z)| \} \tag{17}$$

Note further that since the largest element of A is necessarily positive and lies on the diagonal

$$\max |a_{ij}^{(1)}|_\infty = \max a_{ii}^{(1)} \tag{18}$$

It follows that at most one additional bit is needed in L_S to accommodate this growth since in all practical cases we certainly have $N/2 < K2^{2q_x - 1}$. (In reality it is highly unlikely that this extra bit is ever needed for this growth.)

The effect of these errors on the solution must of course be analysed, too. A larger dynamic range would allow fewer division or scaling operations and so could result in reduced error

bounds.

The wordlength requirements in (19) for the left-hand side matrix during the forward elimination stage follow from (1), the additional bit to allow for growth, and (15):

$$L_S \geq 2 + 2q_X + \log K, \qquad L_A \geq 2L_S + 1 \qquad (19)$$

The growth on the right-hand side is a more serious problem since there is no correspondingly simple bound on the growth factor.

However, we can obtain bounds for the growth of the right-hand side as follows. Corresponding to (14), each stage of the elimination results in modifications of the right-hand side of the form:

$$b_j \leftarrow b_j - b_i \frac{a_{ji}}{a_{ii}} \qquad (20)$$

Now the magnitude of the multiplier here is bounded by the largest matrix element, M_R, say, since the largest element is on the diagonal, so that $|a_{ij}| \leq M_R - 1$ for $i \neq j$, while the diagonal elements are positive integers and so $a_{ii} \geq 1$. Of course, as with the elimination on the left-hand side, the multiplier here would not be computed but rather the multiplication would be performed first and the division would follow.

Denote by M the range available for the initial (scaled) covariance matrix and cross-correlation vector given by the initial lower bound for L_S in (1). That is

$$|a_{ij}| \leq M = K \times 2^{2q_X - 1} \qquad (21)$$

It follows that the growth factor for the right-hand side at each stage is bounded by M so that the real and imaginary parts of the final entry are bounded by M^N which is to say the dynamic range growth is linear in the wordlength. This compares with the faster-than-geometric growth established for the divisionless algorithm in [7].

It follows that any wordlength satisfying

$$L_S \geq N(1 + 2q_X + \log K) + 1 \qquad (22)$$

will certainly suffice. (The additional 1 is again to allow for the very unlikely overspill resulting from the accumulated roundoff error in the divisions.) Such a wordlength will also suffice for the accumulator for the elimination phase since the final multiplication fits into this wordlength, too. The critical dependencies here are clearly on N and q_X. The wordlengths given by (22) are summarized for various cases in Table 1 below. Any technology has a maximum practical integer arithmetic wordlength and this places a restriction on the size of the adaptive beamforming problem which can be solved in this way without some further scaling to restrict the dynamic

range.

TABLE 1

Wordlengths in bits given by (22)

q_x	$K = 2N$			$K = 3N$			$K = 4N$		
	$N = 4$	8	16	4	8	16	4	8	16
4	49	105	225	52	110	235	53	113	241
8	81	169	353	84	174	363	85	177	369
16	145	297	609	148	302	619	149	305	625

The broken line indicates the boundary of what can be achieved with a 256 bit integer wordlength.

It is apparent that increasing the ratio K/N is cheap while increasing either the number of antennas or the quantization is considerably more expensive with the increase in the number of antennas carrying the highest cost in terms of desired wordlength for the elimination phase.

This analysis of dynamic range assumes nothing about the magnitudes of the components of the solution. Any knowledge of the range of these components would lead to much improved dynamic range estimates. This will be considered further in Section 3.3 after we have summarized the corresponding analysis to this for the unscaled case.

3.2   The unscaled case

In the case where the covariance matrix and cross-correlation vectors are not scaled, the initial requirements for the wordlengths are given by (2) and (3). The resulting dynamic range corresponding to (21) will be denoted by

$$|a_{ij}| \leq M' = 2^{1-2q_x} \tag{23}$$

The matrix elements again have a growth factor of 1 and so roundoff effects simply demand one extra bit. The growth analysis for the right-hand side is precisely analogous to that of the previous section and leads to the bound $(M')^N$ from which we can deduce that choosing

$$L_A, L_S \geq N(1 + 2q_x) + 1 \tag{24}$$

will suffice. Table 2 shows the wordlengths given by (24) for the same range of values of N, K and q_x as was used previously. However, in this case there is, of course, no dependence on K.

TABLE 2

Wordlengths in bits given by (24)

q_X	$N = 4$	8	16
4	37	73	145
8	69	137	273
16	133	265	529

Again we see that the wordlengths grow rapidly with both N and q_X, severely limiting the size of problem that can be handled with even a fairly large wordlength integer arithmetic.

To motivate the discussion in the next section, consider the effect of knowing a bound on the weights. If the magnitudes of the weights are known to satisfy some bound of the form

$$\| \boldsymbol{w} \|_1 \leq W \tag{25}$$

then the elements of the right-hand side satisfy

$$| b_j | \leq WM' \tag{26}$$

throughout the elimination phase. A similar conclusion holds for the scaled matrix with M in place of M'. We recall here that the 1-norm $\| \bullet \|_1$ is defined by

$$\| \boldsymbol{z} \|_1 = \sum | z_i | \tag{27}$$

These bounds suggest that much shorter wordlengths may be useable. Information on the size of the right-hand side vector is therefore likely to prove valuable in reducing the computation wordlength requirements for the adaptive beamforming problem.

3.3   "Backward" range analysis

The intention is to compute the weights to an accuracy of q_w bits. These weights are to be represented by the complex integer solution of the linear system. We shall suppose that the calculation of the weights is performed using a wordlength $1+L_w$: that is, to an accuracy of L_w bits together with the sign. Typically, we expect that $L_w > q_w$ to allow for some loss of precision in the solution due to the various roundoff and data errors. L_w therefore represents the appropriate computation wordlength from which weights accurate to q_w bits will be extracted.

Nitzberg [9] studies the precision in the weights that is required in order to achieve a 1 dB or 3 dB degradation as a function of the number N of antennas and the power of the jammer signal. His findings are that q_w depends linearly on the logarithms of both N and SNR_J.

TABLE 3

Weight quantization, q_w, for a 3dB degradation as a function of N and SNR_J from [9]

N	20 dB	30 dB	40 dB	50 dB
4	6	9	13	16
8	7	10	14	17
16	8	11	15	18

We summarize some of Nitzberg's findings in Table 3. For a 1 dB degradation each of these quantization wordlengths should be increased by 1 bit. For example, with $N = 8$ and a 40 dB jammer signal we require 14 bits accuracy in the weights for a 3 dB degradation while 15 bits would be needed for 1 dB. Our present task is to determine from this requirement the appropriate L_w and q_x which will yield this accuracy in the final solution.

In 1980, Nitzberg [10] extended his study to the question of how many bits are needed for the matrix inversion and found a similar linear relation. This work took no account of the formation of the sample covariance matrix or the quantization of the data. To some extent then, the present work is a continuation of Nitzberg's work.

What is the significance of this for our dynamic range analysis? Once L_w is determined, we have a bound on the magnitudes of the weights and therefore can obtain the savings in necessary wordlength anticipated at the end of the previous section. The specific relation between L_w and q_w will depend on the error analysis which follows.

Denote the bound on the weights corresponding to the wordlength L_w by M_w so that

$$| w_j | < M_w = \sqrt{2}\, 2^{L_w} = 2^{L_w - 1/2} \tag{28}$$

From the eigenvalue analysis referred to in Section 2, it follows, using (10) that

$$\| w \|_\infty \leq \| b \|_\infty \tag{29}$$

where b is the unscaled right-hand side vector. Hence using the wordlength given by (2) for the

unscaled matrix it follows that wordlengths satisfying

$$L_w \geq 2q_X + 1/2 \tag{30}$$

will suffice. Now using the bound (28) it follows that

$$\|\mathbf{w}\|_1 \leq NM_w \tag{31}$$

and then using (26) or its equivalent for the scaled matrix, it follows that the modified right-hand side elements generated throughout the solution satisfy

$$|b_j| < NM_wM' \qquad \text{or} \qquad |b_j| < NM_wM \tag{32}$$

With the wordlengths given by (19) for the scaled matrix this yields the wordlength estimate

$$L_S \geq 4q_X + \log N + \log K + 5/2 \tag{33}$$

for the dynamic range of the right-hand side. The corresponding accumulator wordlength need not be sufficient to accommodate a full multiplication of words of this length since the only long multiplies that are needed are between elements of the matrix and the right-hand side vector. It follows that

$$L_A \geq 6q_X + \log N + 2\log K + 4 \tag{34}$$

will suffice.

The only change needed for the unscaled case is that the terms in $\log K$ are not needed:

$$
\begin{aligned}
L_S &\geq 4q_X + \log N + 5/2 \\
L_A &\geq 6q_X + \log N + 4
\end{aligned}
\tag{35}
$$

To illustrate the savings available relative to the earlier tables, we show in Table 4 the resulting wordlengths for the various jammer strengths used in Table 3, with the same combinations of N and K as were used for Tables 1 and 2, with q_X and $L_w \approx 1.5\ q_w$ satisfying (30) and the weight quantizations of Table 3 for a 3 dB degradation.

The linear dependence on q_w (and hence L_w and q_X) and $\log N$ observed by Nitzberg is apparent in these tables. The magnitude of the savings which are obtained from knowing bounds on the weights is also apparent since all of these cases can easily be accommodated by a 128-bit accumulator whereas many of them needed too large a dynamic range for 256 bits in Tables 1 and 2.

TABLE 4

Wordlengths q_X, L_S, L_A given by (30), (33) - (35) with $L_w \approx 1.5 q_w$ taken from Table 3

(a) $SNR_J = 20$ dB

	K = 2N			K = 4N			Unscaled		
N	4	8	16	4	8	16	4	8	16
q_x	4	5	6	4	5	6	4	5	6
L_s	24	30	36	25	31	37	21	26	31
L_A	36	45	54	38	47	56	30	37	44

(b) $SNR_J = 30$ dB

	K = 2N			K = 4N			Unscaled		
N	4	8	16	4	8	16	4	8	16
q_x	7	8	9	7	8	9	7	8	9
L_s	36	42	48	37	43	49	33	38	43
L_A	54	63	72	56	65	74	48	55	62

(c) $SNR_J = 40$ dB

	K = 2N			K = 4N			Unscaled		
N	4	8	16	4	8	16	4	8	16
q_x	10	11	12	10	11	12	10	11	12
L_s	48	54	60	49	55	61	45	50	55
L_A	72	81	90	74	83	92	66	73	80

(d) $SNR_J = 50$ dB

	K = 2N			K = 4N			Unscaled		
N	4	8	16	4	8	16	4	8	16
q_x	12	13	14	12	13	14	12	13	14
L_s	56	62	68	57	63	69	53	58	63
L_A	84	93	102	86	95	104	78	85	92

Clearly the error analysis will be important in determining the dependence of L_w on the desired accuracy q_w which will in turn dictate the actual wordlengths that are needed.

4. Back substitution

To determine the dynamic range requirements for the back substitution phase of the solution, we can consider the two cases (the scaled and unscaled covariance matrix) together since the rest of the analysis is similar for both. At this stage we are interested in the solution of a system

$$U\mathbf{w} = \mathbf{b} \tag{36}$$

where U is an upper triangular matrix.

Using the notation of the previous section, we have the following bounds for the elements of this system:

elements of	scaled	unscaled
U	M	M'
b	NM_wM	NM_wM'
w	M_w	M_w

In fact, for the individual components of the right-hand side vector b, we can obtain the tighter bounds:

$$|b_i| \leq (N+1-i)M_w(M \text{ or } M') \tag{37}$$

for the same two cases.

The bounds for the final weights clearly can be accommodated in the same wordlengths that were used for the forward elimination phase. The only point of concern is therefore the accumulation of the (modified) right-hand side prior to each division in the standard loop:

Back substitution algorithm

$$w_N := b_N / u_{NN}$$

for i = N-1 down to 1

$$w_i := \left[b_i - \sum_{j=i-1}^{N} u_{ij}w_j \right] \Big/ u_{ii}$$

Note again that each division operation has a real divisor so that complex division is avoided. To see that the same accumulator that was required for the elimination suffices for this stage of the solution, we must consider the right-hand side of the above loop operation.

Temporarily, we denote by B the quantity M_wM or M_wM' whichever is appropriate to the scaling being used. Since the final value of each w_i is bounded by M_w it follows that the final accumulated value before the division is bounded by B. Its component parts, using (37), satisfy

$$|b_i| \leq (N+1-i)B, \qquad \left| \sum_{j=i-1}^{N} u_{ij}w_j \right| \leq (N-i)B \qquad (38)$$

from which it follows that no partial result can exceed $(N+1-i)B$ and therefore that the accumulator lengths derived in the previous section suffice. Furthermore, with such an accumulator, the only error committed is the rounding in storing the result of the final division.

The wordlengths of Table 4 are suitable for the complete solution process for Gauss elimination using integer arithmetic with divisions in such a way that correct integer results are accumulated throughout the process with correctly rounded integer results for division. The purpose of the analysis which follows is to analyse this particular form of Gauss elimination to determine the data quantization and arithmetic capability which are required in order to deliver a specified accuracy in the final weights. In the next section we consider the error analysis aspects of this question but, first, we complete the analysis of the dynamic range requirements.

Subsequently, we will address similar questions for the situation where some scaling is necessary in order to restrict the dynamic range to keep the accumulator size below some threshold value. For example, speed considerations may dictate that arithmetic wordlengths are kept below 32 or 64 bits.

There is, in fact, an even smaller bound available for the right-hand side than that given by (38). This derives from the fact that the first equation remains unchanged during the elimination so that b_1 is bounded by the original M or M'. Similarly the second element cannot undergo the full growth anticipated here and can only achieve a magnitude close to M^2 or M'^2. In summary, the factor of N in the bounds (32) can be replaced by $N-2$. However this represents a saving of only 1 bit in the $N = 4$ case and even less than that for larger values of N. For this reason the overall wordlengths in Table 4 should be used.

It is apparent that the wordlengths required are much more moderate than was predicted in Tables 1 and 2 even though the data quantization wordlengths have a similar range to those used there. Nonetheless, an accumulator length limit of even 64 bits would place real restrictions on the sizes of problems to be solved. A smaller limit would clearly be very restrictive without some scaling of the right-hand side vector during the computation. Only the right-hand side would need scaling

since the matrix elements are not subject to growth except as a result of rounding errors. This suggests another possibility: different quantization for the data matrix and the desired response to keep the dynamic range for the right-hand side smaller. Such a trade-off may be considered later.

5. Error analysis

We must consider the effect of errors from a variety of sources:

quantization errors in the data matrix and the desired response,

the formation of the covariance matrix and cross-correlation vector, and

rounding errors in the divisions in the elimination and back substitution phases.

Conventional error analyses can be used for some parts of this but the fixed-point arithmetic with extended accumulator that was discussed previously does not lend itself immediately to those analyses which are liable to produce overly pessimistic results in this case.

The first-order effect of the propagation of the data errors can be modeled on conventional analyses such as those of Wilkinson [14], [13] but the results which are included below for completeness are inappropriate for our integer arithmetic if the errors are such that second order effects are truly negligible.

Because of the "integerized" nature of the data matrix and desired response, the real and imaginary parts of the quantization errors are each bounded by 1/2 so that the error in any element of the data matrix is bounded by $1/\sqrt{2}$. In this section, we denote the *computed* covariance matrix and cross-correlation vector by A and $\mathbf{b}$ respectively to distinguish these from their theoretical counterparts. We shall also denote the computed solution for the weights by $\hat{\mathbf{w}}$.

First, we consider the effects of the quantization errors on the computed solution. Here and throughout this section we shall only consider first-order effects. The elements of the data matrix are quantized to q_X-bit complex integers which (assuming correct rounding) have errors $\leq 1/2$ in both their real and imaginary parts. Hence the errors in the data matrix are bounded by:

$$|\delta x_{ij}| \leq 1/\sqrt{2} \tag{39}$$

Elements of the (scaled) covariance matrix are formed from inner products of the snapshot

vectors. A product of two such numbers u, v say, computed in exact integer arithmetic has an error bounded by

$$|\delta(uv)| = |u\delta v + v\delta u + \delta u\delta v| \leq \frac{|u| + |v|}{\sqrt{2}} \leq 2^{q_x-1} \tag{40}$$

neglecting the second-order term. It follows that the computed elements of the scaled covariance matrix have errors bounded by

$$|\delta a_{ij}| \leq K \times 2^{q_x-1} = \frac{M}{\sqrt{2}\,2^{q_x}} \tag{41}$$

which is equivalent to the statement that (at least) the first q_x bits of each element are correct. This leads to a third natural possible scaling of the original problem in which the matrix and right-hand side would be stored to this accuracy. This gives rise to a modified error and range analysis which will be considered later.

From (41), it follows that

$$\|\delta A\|_\infty \leq NK2^{q_x+1} \tag{42}$$

and, similarly,

$$\|\delta b\|_\infty \leq K2^{q_x-1} \tag{43}$$

For the true (unscaled) covariance matrix, the rounding errors resulting from the division by K is of similar magnitude to the already neglected second-order error term and so there are bounds similar to those in (41) - (43) except that the factor K is not present.

To estimate the effect of these errors on the computed solution, we use a first-order analysis which is a slight modification of the usual Wilkinson-style relative error analysis to this situation. With no arithmetic errors during the solution process, the computed solution satisfies

$$(A + \delta A)\hat{w} = b + \delta b = Aw + \delta b \tag{44}$$

from which we obtain the following error "bound" which is dependent on the computed solution:

$$\|\hat{w} - w\|_\infty \leq \|A^{-1}\|_\infty (\|\delta A\|_\infty \|\hat{w}\|_\infty + \|\delta b\|_\infty) \tag{45}$$

Using the first-order estimate:

$$\|\hat{w}\|_\infty \approx \|w\|_\infty = \|A^{-1}b\|_\infty \leq \|A^{-1}\|_\infty \|b\|_\infty \tag{46}$$

and using Compton's eigenvalue analysis, we see from (10) that $\lambda_{min} \geq 1$ so that $\|A^{-1}\|_\infty \leq 1$. Substituting this in (46) and (45), we obtain the bounds

$$\| \delta \mathbf{w} \|_\infty = \| \hat{\mathbf{w}} - \mathbf{w} \|_\infty \leq \| \delta A \|_\infty \| \mathbf{w} \|_\infty + \| \delta \mathbf{b} \|_\infty$$
$$\leq \| \delta A \|_\infty \| \mathbf{b} \|_\infty + \| \delta \mathbf{b} \|_\infty \tag{47}$$

which indicate that the error bound is of the same order of magnitude as the solution itself. Such a bound is not useful.

Wilkinson [13] also includes a summary of the corresponding fixed-point analysis for the situation where the correct binary exponent of all elements of the solution is known - *and* is constant over the weight vector. Such an analysis is not immediately applicable in our situation but its basic principles can be used here if we make the one additional assumption that the magnitudes of the weights (both true and computed) are such that second order error effects can be neglected. We shall make this assumption throughout the remainder of this section.

Since we are computing integer solutions using longer wordlengths than are strictly necessary this is not much more than the assumption that all weights are nonzero which in turn is equivalent only to saying that there is no redundancy in the antenna array.

From our earlier analysis, (41) and (43), we know that the maximum error in elements of the (scaled) covariance matrix and cross-correlation vector is bounded by

$$M / \left(\sqrt{2} \, 2^{q_x} \right) = K 2^{q_x - 1} =: E \tag{48}$$

say. There is a corresponding error bound for the unscaled case:

$$M' / \left(\sqrt{2} \, 2^{q_x} \right) = 2^{q_x + 1} =: E' \tag{49}$$

Wilkinson's [13] pp111-2 fixed point error analysis can be modified to our situation by regarding the various integer quantities as fixed point fractions of some global bound. The wordlengths chosen are then sufficient for exact accumulation of scalar products and for the use of this "long accumulator" for division.

With this interpretation, it also follows that the assumption [13] p212 that all matrix elements remain bounded by unity throughout the solution process is valid in this case. This follows from the "no-growth" result summarized in (16). In turn this implies that results corresponding to those of [13] pp 209-11 are valid for our system and arithmetic.

From (41), we have already observed that the leading q_X bits of all matrix elements are correct. Neglecting any second-order effects and recalling that, because of the greater accumulator lengths discussed above, any inner products and divisions can be formed using a "long" accumulator then there is maximum error in elements of the upper triangular factor and corresponding right-hand side, *regarded as fractions*, of $(N - 1) 2^{-q_X}$.

The dynamic range analysis for back substitution in the last section establishes that the inner products formed during this phase can be computed exactly. Compare Section 11 of [14] for the situation where the order of magnitude of the components of the solution is known. The critical feature of that analysis is that the magnitudes of the roundoff errors are then determined by the working precision or wordlength. Although the corresponding order of magnitude is neither fixed nor known here, the dynamic range established in the preceding sections implies knowledge of the magnitude of roundoff errors.

Again interpreting all our integers as fixed-point fractions, it then follows that the back substitution therefore results in a further error bounded by 2^{-q_X}. Neglecting any second-order effects, it follows that the final computed solution has components with errors bounded by NE or $N 2^{-q_X}$. Such an error corresponds to a further loss in precision of at most $\log N$ bits in the real and imaginary parts of the weights. Thus we require that q_X be at least this much greater than the data quantization q_w, that is

$$q_w \geq q_X + \log N \tag{50}$$

from which using (30) - (35) it follows that

$$L_w \geq 2 (q_w + \log N) + 1$$

$$L_S \geq 4 q_w + 5 \log N + 5/2 \quad (+ \log K) \tag{51}$$

$$L_A \geq 6 q_w + 7 \log N + 4 \quad (+ 2 \log K)$$

where the final parenthetic terms are included in the scaled case.

In every case in Table 3, this yields a value for $L_w > 1.5 q_w$ so that the wordlengths derived in Table 4 are inadequate for this process. We use (50) and (51) to get the revised wordlengths

shown in Table 5.

TABLE 5

Wordlengths q_X, L_S, L_A given by (50) and (51) with q_w taken from Table 3

(a) $SNR_J = 20$ dB

	$K = 2N$			$K = 4N$			Unscaled		
N	4	8	16	4	8	16	4	8	16
q_X	8	10	12	8	10	12	8	10	12
L_S	40	50	60	41	51	61	37	46	55
L_A	60	75	90	62	77	92	54	67	80

(b) $SNR_J = 30$ dB

	$K = 2N$			$K = 4N$			Unscaled		
N	4	8	16	4	8	16	4	8	16
q_X	11	13	15	11	13	15	11	13	15
L_S	52	62	72	53	63	73	49	58	67
L_A	78	93	108	80	95	110	72	85	98

(c) $SNR_J = 40$ dB

	$K = 2N$			$K = 4N$			Unscaled		
N	4	8	16	4	8	16	4	8	16
q_X	15	17	19	15	17	19	15	17	19
L_S	68	78	88	69	79	89	65	74	83
L_A	102	117	132	104	119	134	96	109	122

(d) $SNR_J = 50$ dB

	$K = 2N$			$K = 4N$			Unscaled		
N	4	8	16	4	8	16	4	8	16
q_X	18	20	22	18	20	22	18	20	22
L_S	80	90	100	81	91	101	77	86	95
L_A	120	135	150	122	137	152	114	127	140

We note that these wordlengths are such that even a 128-bit processor is inadequate for many problems. This suggests that some scaling would be necessary in order to keep wordlengths to

a practical level.

One natural approach to this which should be pursued is to take advantage of the fact that only q_x bits of the initial matrix and right-hand side are correct and so store only these. This is equivalent to a scaling of the linear system which halves the initial wordlengths from which the growth takes place. Of course this does not simply mean that all subsequent wordlengths are halved and both the dynamic range and error analysis needs to be reworked for this situation.

Conclusions

In this paper, we have derived equations to allow trade-offs between word size for the adaptive weights, data quantization, dynamic range (word length), and accumulator word size, for the conventional Gauss Elimination algorithm, using an integer processor. We have found that a very large word length is required for a moderately sized adaptive beamforming problem. It is obvious that for large problems, say greater than 16 antenna elements, that scaling is required to keep the word size down. This scaling will degrade the accuracy of the adaptive weights.

This problem must be examined further to validate the practicality of using an integer (e.g. RNS) processor. In a subsequent paper, we will study the trade-offs for the divisionless Gauss Elimination algorithm of Kirsch and Turner or QR-Decomposition, using scaling instead of division in the conventional implementation.

References

[1]   R.T.Compton *Adaptive Antennas: Concepts and Performance*, Prentice Hall, 1988

[2]   J.W.Demmel, *Trading off parallelism and numerical stability*, pp. 49-68 in Linear Algebra for large-scale and real-time applications (G.Golub, M.Moonen and B.de Moor, eds.) Kluwer, 1993

[3]   G.H.Golub and C.F.Van Loan, *Matrix Computations*, Johns Hopkins Press, Baltimore, 1989.

[4]   M. Griffin, M. Sousa and F.J. Taylor, *Efficient scaling in the residue number system*, Proc IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing, IEEE, New York, 1989

[5]   S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 1986

[6]   B.J.Kirsch and P.R.Turner, *Adaptive beamforming using RNS arithmetic*, pp 36-43, ARITH11, (11th Symp Computer Arithmetic) IEEE Computer Society, Washington DC, 1993.

[7]   B.J.Kirsch and P.R.Turner, *Modified Gaussian elimination for adaptive beamforming using RNS arithmetic*, NAWC-AD Tech Report 94112-50, 1994.

[8]   R.A.Monzingo and T.W.Miller *Introduction to Adaptive Arrays*, Wiley-Interscience, 1980.

[9]   R. Nitzberg, *Effect of errors in adaptive weights,* IEEE Trans AES 12 (1976) 369-373.

[10]   R. Nitzberg, *Computational precision requirements for optimal weights in adaptive processing*, IEEE Trans AES 16 (1980) 418-425.

[11]   M.A. Soderstrand, W.K. Jenkins, G.A. Jullien, and F.J. Taylor, *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*, IEEE, New York, 1986.

[12]   C.B.Ward, P.J.Hargrave and J.G.McWhirter, *A novel algorithm and architecture for adaptive digital beamforming*, IEEE Trans Antennas&Propagation 34 (1986) 338-346.

[13]   J.H.Wilkinson, *Error analysis of direct methods of matrix inversion*, J. ACM 8 (1961) 281-330.

[14]   J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, 1965.

DISTRIBUTION LIST

US NAVAL ACADEMY...10
ANNAPOLIS, MD 21402
  (ATTN: MATHEMATICS DEPARTMENT, PETER R. TURNER)

AVIONICS DEPARTMENT..12
ENGINEERING DIVISION (CODE 4.5.5.1)
NAVAL AIR WARFARE CENTER
AIRCRAFT DIVISION WARMINSTER
P. O. BOX 5152
WARMINSTER, PA 18974-0591)
  (10 FOR CODE 4.5.5.1, BARRY J. KIRSCH)
  (2 FOR CODE 7.2.5.5)

DEFENSE TECHNICAL INFORMATION CENTER..........................2
ATTN: DTIC-FDAB
CAMERON STATION BG5
ALEXANDRIA, VA 22304-6145

CENTER FOR NAVAL ANALYSIS.....................................1
4401 FORT AVENUE
P. O. BOX 16268
ALEXANDRIA, VA 22302-0268

OFFICE OF NAVAL RESEARCH......................................2
800 NORTH QUINCY STREET
ARLINGTON, VA 22217-5660
  (2 FOR ONR-313)